BAB II
LANDASAN TEORI

Pada bagian ini akan dijelaskan mengenai beberapa teori penunjang yang berhubungan dengan pokok bahasan dalam mendukung penelitian sistem survey untuk pendamping PKH pada tugas akhir ini. Adapun pokok-pokok penelitian yang dibahas adalah dinas sosial, buku panduan pendamping PKH, andorid, bahasa pemrograman android, web dan basis data (*database*).

2.1 Dinas sosial

Dinas sosial merupakan pelaksanaan otonomi daerah pada bidang sosial tenaga kerja, yang mempunyai fungsi melaksakan dan melakukan pembinaan pada bidang penyuluhan dan bimbingan sosial yang terkait, untuk memberikan suatu bentuk motivasi, memonitoring dan konsultasi pada setiap masyarakat yang membutuhkan[14].

2.2 Program Keluarga Harapan (PKH)

PKH adalah program perlindungan sosial yang memberikan bantuan kepada Rumah Tangga Sangat Miskin (RTSM) dan bagi anggota keluarga diwajibkan melaksanakan persyaratan dan ketentuan yang telah ditetapkan di bidang pendidikan dan kesehatan[3].

Landasan Hukum pemberian PKH adalah:

1. Undang-undang nomor 40 Tahun 2004 tentang Sistem Jaminan Sosial Nasional.
2. Undang-undang nomor 13 Tahun 2011 tentang penanganan Fakir Miskin.
3. Peraturan Presiden nomor 15 Tahun 2010 tentang Percepatan Penanggulangan Kemiskinan.
4. Inpres nomor 3 Tahun 2010 tentang Program Pembangunan yang Berkeadilan poin lampiran ke 1 tentang Penyempurnaan Pelaksanaan Program Keluarga Harapan.
5. Inpres nomor 1 Tahun 2013 tentang Pencegahan dan Pemberantasan Korupsi poin lampiran ke 46 tentang Pelaksanaan Transparansi Penyaluran Bantuan

Langsung Tunai Bersyarat Bagi Keluarga Sangat Miskin (KSM) Sebagai Peserta Program Keluarga Harapan (PKH).

Dasar Pelaksanaan PKH

1. Keputusan Menteri Koordinator Bidang Kesejahteraan Rakyat selaku ketua Tim Koordinasi Penanggulangan Kemiskinan, No: 31/KEP/MENKO/-KESRA/IX/2007 tentang "Tim Pengendali Program Keluarga Harapan" tanggal 21 September 2007
2. Keputusan Menteri Sosial Republik Indonesia No. 02A/HUK/2008 tentang "Tim Pelaksana Program Keluarga Harapan (PKH) Tahun 2008" tanggal 08 Januari 2008.
3. Keputusan Gubernur tentang "Tim Koordinasi Teknis Program Keluarga Harapan (PKH) Provinsi/TKPKD".
4. Keputusan Bupati/Walikota tentang "Tim Koordinasi Teknis Program Keluarga Harapan (PKH) Kabupaten/Kota/TKPKD".
5. Surat Kesepakatan Bupati untuk Berpartisipasi dalam Program Keluarga Harapan.

2.2.1 Tujuan PKH

Tujuan umum PKH yaitu untuk meningkatkan kualitas sumber daya manusia, mengubah pandangan semua calon peserta PKH terhadap upaya pemerintah untuk peningkatan kesejahteraan, dan diharapkan dapat mengurangi rantai kemiskinan antar generasi yang terus berkembang.

Secara khusus tujuan PKH adalah sebagai berikut[10]:

1. Meningkatkan kualitas dan mutu kesehatan Keluarga Sangat Miskin (KSM)
2. Meningkatkan kualitas dan mutu pendidikan anak- anak KSM
3. Meningkatkan semua akses dan kualitas pada pelayanan pendidikan dan kesehatan, khususnya bagi mereka anak - anak KSM

2.2.2 Syarat kepesertaan PKH

Syarat ketentuan kepesertaan PKH adalah diutamakan bagi para keluarga sangat miskin (KSM) yang sudah terbukti tersurvey oleh para pendamping PKH yang selanjutnya telah mendapatkan persetujuan sebagai peserta PKH serta sudah ditetapkan oleh Kementrian Sosial[10].

2.2.3 Komponen PKH

Komponen PKH terdiri dari[10]:

1. Ibu Hamil/Nifas
2. Anak usia dibawah lima tahun (Balita)
3. Anak SD dan yang sederajat
4. Anak SMP dan yang sederajat
5. Anak SMA dan yang sederajat
6. Anak usia pra sekolah
7. Anak Penyandang disabilitas

2.2.4 Penerima bantuan PKH

Penerima bantuan PKH adalah diutamakan bagi para Keluarga Sangat Miskin (KSM) dan peserta PKH yang sudah mendapatkan persetujuan sebagai peserta PKH, yaitu[10]:

1. Ibu hamil/ibu nifas/ anak balita
2. Anak berusia kurang dari 7 tahun yang belum masuk pendidikan dasar (anak pra-sekolah)
3. Anak usia 7-21 tahun yang belum menyelesaikan pendidikan wajib belajar 12 tahun
4. Anak penyandang dusabilitas berusia 0-21 tahun.

2.3 Metode Perancangan Sistem

Metode perancangan yang digunakan pada perancangan aplikasi Mobile Survey Pendamping Program Keluarga Harapan (PKH) ini adalah metode pengembangan sistem SDLC (*System Development Life Cycle*) model prototype.

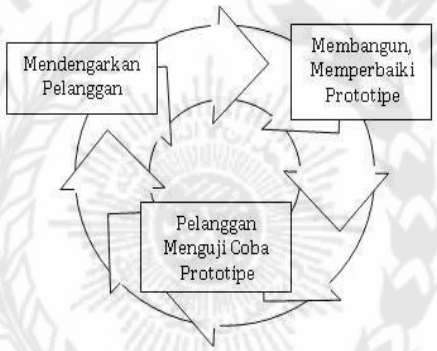
2.3.1 SDLC (*System Development Life Cycle*)

SDLC (*System Development Life Cycle*) merupakan proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model - model atau metodologi yang digunakan untuk mengembangkan sistem - sistem perangkat lunak sebelumnya (berdasarkan *best pratice* atau cara - cara yang sudah teruji baik). Terdapat banyak model yang dapat digunakan dalam membangun maupun mengembangkan sistem berbasis

SDLC yaitu: Model *Waterfall*, Model *Prototype*, Model *Rapid Application Development* (RAD), Model *Iterative* dan *Spiral*[15].

2.3.2 SDLC (*System Development Life Cycle*) Model *Prototype*[15]

Model pengembangan sistem *prototype* sangat baik digunakan untuk menyelesaikan masalah dalam kesalahpahaman antara *user* dan *analis* yang timbul akibat *user* tidak mampu mendefenisikan secara jelas kebutuhannya. *Prototyping* adalah pengembangan yang cepat dan pengujian terhadap model kerja (prototipe) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem dan ahli bisnis. *Prototyping* disebut juga *Rapid Application Design* (RAD) karena menyederhanakan dan mempercepat desain sistem.



Gambar 2.1 SDLC Model *Prototype* Sumber: [15]

Berikut ini merupakan penjelasan dari tahapan - tahapan yang dilakukan pada saat mengembangkan sistem dengan menggunakan model *prototype*:

1. Mendengarkan Pelanggan

   Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.

2. Merancang dan Membuat *Prototype*

   Pada tahap ini, dilakukan perancangan dan pembuatan prototipe sistem yang disesuai dengan kebutuhan pengguna.

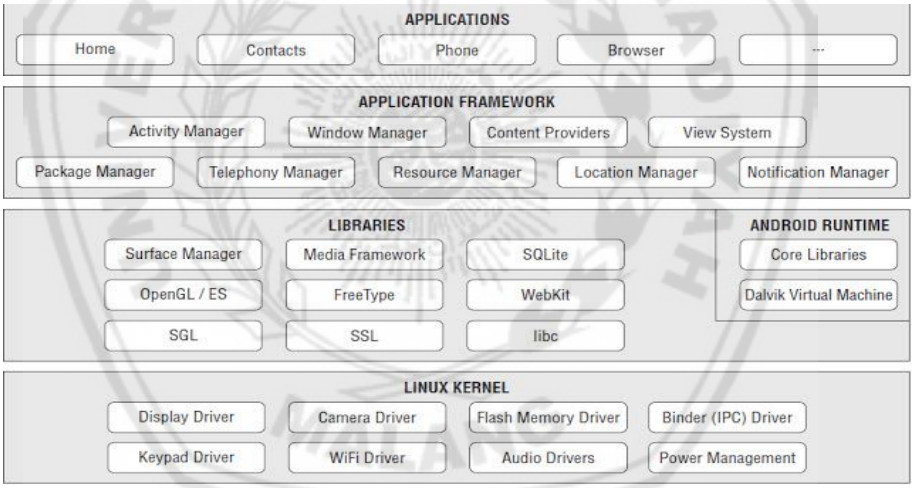
3. Uji Coba

Pada tahap ini, dilakukan pengujian *prototype* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan prototipe, maka sistem akan diselesaikan sepenuhnya. Namun, jika masih belum sesuai kembali ke tahap pertama.

2.4 Android

Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (*touch screen*) yang berbasis linux[4].

Google mengibaratkan Android sebagai sebuah tumpukan *software*, dimana setiap tumpukan berisi program yang mendukung fungsi-fungsi spesifik dari sistem operasi[4]. Berikut ini susunan lapisan - lapisan tersebut jika dilihat dari dasar hingga lapirasan teratas adalah sebagai berikut



Gambar 2.2 Arsitektur Aplikasi Android Sumber: [4]

2.4.1 *Linux* sebagi *Kernel*

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup *memory management*, *security setting*, *power management*, dan beberapa *driver hardware*. Kernel berperan sebagai *abstraction layer* antara *hardware* dan keseluruhan *software*. Sebagai contoh, HTC GI dilengkapi dengan kamera yang memungkinkan pengguna mengirimkan perintah ke *hardware* kamera[8].

2.4.2 *Android Runtime*

Lapisan setelah Kernel Linux adalah Android *Runtime*. Android *Runtime* ini berisi *Core Libraries* dan *Dalvik Virtual Machine*. *Core Libraries* mencakup serangkaian inti *library* Java, artinya Android menyertakan satu set *library-library* dasar bahasa pemrograman Java[8].

Dalvik adalah Java *Virtual Machine* yang memberi kekuatan pada sistem Android. *Dalvik* VM ini di optimalkan untuk telepon seluler.

Setiap aplikasi yang berjalan pada Android berjalan pada processnya sendiri, dengan *instance* dari *Dalvik Virtual Machine*. *Dalvik* telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi *Virtual Machine* dengan efisien. *Dalvik* VM dapat mengeksekusi file dengan format *Dalvick Executable* (.dex) yang telah dioptimasi untuk menggunakan minimal memory *footprint Virtual Machine* ini *register-based,* dan menjalankan *class-class* yang di *compile* menggunakan *compiler* Java yang kemudian ditransformasi menjadi format (.dex) menggunakan "dx" *tool* yang telah disertakan.

Dalvik Virtual Machine (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti *threading* dan *low-level memory management.*

2.4.3 *Libraries*

Bertempat di level yang sama dengan Android Runtime adalah *Libraries*. Android menyertakan satu set *library-library* dalam bahasa C/C++ yang digunakan oleh bergbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh *programmer* melewati Android *application framework*. Sebagai contoh Android mendukung pemutaran audio, video, dan gambar[8].

2.4.4 *Aplication framework*

Lapisan selanjutnya adalah *Aplication Framework* yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*. *Application Framework* adalah serangkaian tool dasar seperti alokasi *resource smartphone*, aplikasi telepon, pergantian antar proses atau program, dan

pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada *tool-tool* dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks[8].

Programmer mendapatkan akses penuh untuk memanfaatkan API-API (*Android Protocol Interface*) yang juga digunakan *core applications*. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen-komponen setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama memungkinkan pengguna mengganti komponen-komponen yang dikehendaki.

Di dalam semua aplikasi terdapat *service* dan sistem yang meliputi:

1. Satu *set Views* yang dapat digunakan untuk membangun aplikasi meliputi *lists, grids, text boxes, buttons*, dan *embeddable web browser*
2. *Content Providers* yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya *Contacts*), atau untuk membagi data yang dimilikinya.
3. *Resource Manager*, menyediakan akses ke *non-code resources* misalnya *localized strings, graphics*, dan *layout files.*
4. *Notification Manager* yang memungkinkan semua aplikasi untuk menampilkan *custom alerts* pada status bar.
5. *Activity Manager* yang memanage *life cycle* of dari aplikasi dan menyediakan *common navigation backstack*

2.4.5 *Application*

Di lapisan teratas bercokol aplikasi itu sendiri. Di lapisan inilah anda menemukan fungsi-fungsi dasar *smartphone* seperti menelepon dan mengirim pesan singkat, menjalankan *web browser*, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi-fungsi dasar tersebut melalui *user interface*[8].

2.5 *JSON*

JSON (*JavaScript Object Notation*) merupakan format pergantian data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diproses dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari

Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, *Java*, *JavaScript*, *Perl*, *Python* dan lain - lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

Kelebihan format data JSON adalah lebih unggul dibandingkan XML mulai dari kecepatan, penulisan yang lebih gampang dan coding untuk parsing yang lebih ringkas dan sederhana[8].

2.5.1 Struktur Penulisan JSON

Terdapat 2 tanda penting dalam penulisan json, yaitu { } dan [ ]. Tanda kurung kurawal ({ }) menandakan *JSONObject* dan tanda kurung kotak ([ ]) menandakan *JSON Array*.

JSON terbuat dari dua struktur:

1) Kumpulan pasangan nama/nilai,. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), table hash (*hash table*), daftar berkunci (*keyed list*) atau *associative array*.
2) Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vector (*vector*), daftar (*list*) atau urutan (*sequence*).

JSON menggunakan bentuk sebagai berikut:

1) **Object** adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan kurung kurawal buka ({) dan diakhiri dengan kurung kurawal tutup (}). Setiap nama diikuti dengan titik dua (:) dan setiap pasangan nama/nilai dipisahkan oleh koma (,).
2) **Array** adalah kumpulan nilai yang terurutkan. Array dimulai dengan kurung kotak buka ([ ) dan diakhiri dengan kurung kotak tutup ( ]). Setiap nilai dipisahkan oleh tanda koma (,).
3) **Nilai** (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah *object* atau sebuah larik.
4) **String** adalah kumpulan dari nol atau lebih karakter *Unicode*, yang

dibungkus dengan tanda kutip ganda. Di dalam *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.

5) **Angka** sangat mirip dengan angka di C atau Java, kecuali format *octal* dan heksadesimal tidak digunakan[8].

2.6 *Aplikasi* Berbasis Web

Aplikasi web (*web application*) merupakan suatu aplikasi yang diakses menggunakan suatu browser web melalui suatu jaringan internet atau intranet. Dan juga merupakan suatu aplikasi perangkat lunak komputer yang dikhususkan dalam bahasa yang didukung oleh browser web (seperti HTML, *JavaScript*, AJAX, *Java*, dll) dan sudah terhubungkan pada browser tersebut untuk menampilkan aplikasi yang dibuat.

Aplikasi *web* menjadi sangat populer dikarena kemudahan dalam memberikan ketersedianya untuk mengaksesan dari para client, browser web, yang kadang disebut sebagai suatu *thin client* (workstatio / perangkat kecil komputer). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada sebuah komputer klien merupakan alasan kunci popularitasnya hanya dengan mengakses dari web browser aplikasi dapat digunakan[7].

2.7 *Framework CodeIgniter*

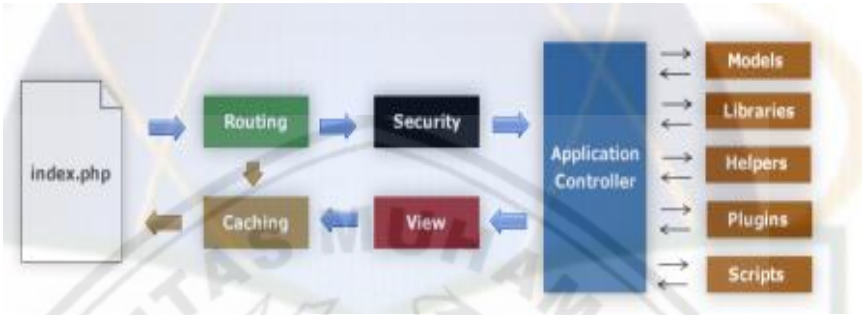
CodeIgniter adalah sebuah framework PHP. Framework itu sendiri adalah suatu kerangkakerja yang berupa sekumpulan folder yang memuat file - file php yang menyediakan class *libraries*, *helpers*, *plugins* dan lainnya . *Framework* menyediakan konfigurasi dan teknik coding tertentu [7].

CodeIgniter adalah sebuah web *application framework* yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan *Codeigniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. *Codeigniter* menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. CodeIgniter diperkenalkan kepada publik pada tanggal

28 februari 2006. *CodeIgniter* sendiri merupakan salah satu framwork tercepat dibandingkan dengan *framework* lainnya[7].

CodeIgniter sendiri dibangun menggunakan konsep *Model - View - Controller development pattern.*

Untuk mengetahui lebih jelas tentang suatu proses alur kerja dari sistem yang menggunakan *CodeIgniter Framework* dapat dilihat pada **Gambar 2.3** sebagai berikut[7].



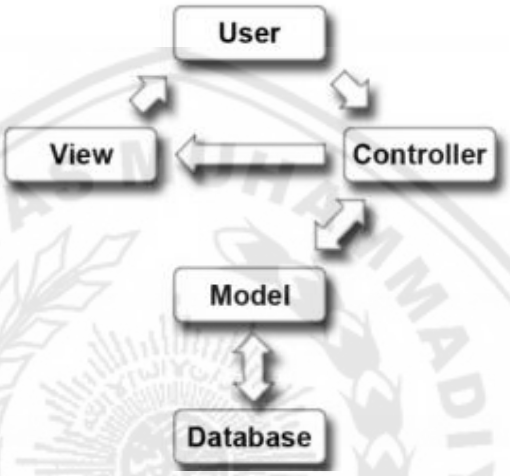
Gambar 2.3 Alur kerja CodeIgniter *Framework,* Sumber: [7]

Keterangan:

1. **Index.php** berfungsi sebagai *front controller*, menginisialisai *base resource* untuk menjalankan *CodeIgniter*.
2. **Router** memeriksa HTTP *request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika **Cache** aktif, maka hasilnya akan langhsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. **Security** sebelum *controller* dimuat, HTTP *request* dan data yang dikirimkan *user* akan difilter untuk keamanan.
5. **Controller** memuaut *model, core libraries, plugins, helpers,* dan semua *resource* yang diperlukan untuk memproses data *request*.
6. **View** yang dihasilkan akan dikirim ke *browser*, jika *cache* aktif, maka *view* akan disimpan sebagai *cache* dahulu, sehingga pada *request* berikutnya langsung ditampilkan.

2.8 *Model Veiw Controller (MVC)*

Model-View-Controller (MVC) adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter . MVC adalah singkatan dari *Model View Controller*. MVC sebenarnya adalah sebuah pattern / teknik pemogramanan yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain:



Gambar 2.4 Cara Kerja *MVC (Model View Controller)*, Sumber: [7].

Berdasarkan gambar 2.3 diatas dapat kita ketahui bahwa ketika datang sebuah permintaan dari *user*, maka permintaan tersebut akan ditangani oleh *Controller*, kemudian *Controler* akan memanggil *Model* jika memang diperlukan operasi database. Hasil *query* pada *Model* kemudian dikembalikan ke *Controller*, Selanjutnya *Controller* yang akan memanggil *View* pada saat yang tepat dan mengkombinasikannya dengan hasil *queri Model* tersebut. Hasil akhir dari operasi ini akan ditampilkan ke *browser* yang selanjutnya bisa dilihat oleh *user*.

2.8.1 *Model*

Model berhubungan dengan data dan interaksi ke *database* atau webservice. *Model* juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun *webservice*[7].

2.8.2 *View*

View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, *javascript* dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di *view*. Di dalam *view* hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. *View* dapat dikatakan sebagai halaman *website* yang dibuat dengan menggunakan HTML dan bantuan CSS atau *JavaScript*[7].

2.8.3 *Controller*

Controller bertindak sebagai penghubung data dan view. Di dalam *Controller* inilah terdapat *class-class* dan fungsi-fungsi yang memproses permintaan dari *View* ke dalam struktur data di dalam Model. Tugas *controller* adalah menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/*error*, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input[7].

2.9 *MySQL*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta *instalasi* di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL

Tidak sama dengan proyek-proyek seperti *Apache*, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius[11].